

#4
500.41130X00



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Jun YOSHIDA et al
Serial No.: NOT YET ASSIGNED
Filed: January 30, 2002
Title: PROGRAM CONTROL, COMPUTER SYSTEM, CONTROL
PROGRAM, AND STORAGE MEDIUM STORING THE
CONTROL PROGRAM
Group: NOT YET ASSIGNED

LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

January 30, 2002


Sir:

Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s)
hereby claim(s) the right of priority based on Japanese Patent Application No.(s)
2001-328505, filed October 26, 2001.

A certified copy of said Japanese Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP



Carl I. Brundidge
Registration No. 29,621

CIB/nac
Attachment
(703) 312-6600

日 本 国 特 許 庁
JAPAN PATENT OFFICE

J1000 U.S. PRO
10/058780
01/30/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年10月26日

出 願 番 号

Application Number:

特願2001-328505

[ST.10/C]:

[JP2001-328505]

出 願 人

Applicant(s):

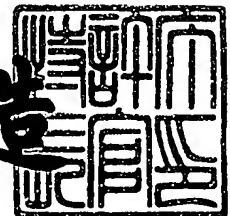
株式会社日立製作所

CERTIFIED COPY OF
PRIORITY DOCUMENT

2002年 1月11日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3114284

【書類名】 特許願

【整理番号】 K01009351A

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/00

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション事業部内

 【氏名】 吉田 順

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション事業部内

 【氏名】 橋本 哲也

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション事業部内

 【氏名】 山本 展之

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション事業部内

 【氏名】 山本 祐史

【発明者】

 【住所又は居所】 神奈川県川崎市幸区鹿島田 8 9 0 番地 株式会社日立製作所 ビジネスソリューション事業部内

 【氏名】 紅山 伸夫

【特許出願人】

 【識別番号】 000005108

 【氏名又は名称】 株式会社日立製作所

【代理人】

 【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム制御方法および計算機システム並びに制御プログラムおよびそのプログラムを格納した記録媒体

【特許請求の範囲】

【請求項 1】

第一のプログラムから出力されたデータを入力として第二のプログラムの入力可能な形式に変換して第二のプログラムに出力するアダプタプログラムを有するプログラム制御方法において、

上記アダプタプログラムは、第一のプログラムからデータを含む起動要求を入力するステップと、起動されたアダプタ実行部における複数の業務で共通に利用するアダプタコンポーネントを呼び出すためのアダプタ呼び出し定義情報を参照してアダプタコンポーネント参照名を取得するステップと、前記アダプタ呼び出し定義情報を参照して業務識別情報を取得するステップと、所定のアダプタコンポーネントのインタフェースに整合した形式にデータを変換するステップと、業務識別情報と前記変換データをアダプタコンポーネントに渡すステップと、業務識別情報をもとに第二のプログラムを呼び出すためのコネクタ呼び出し定義情報を選択するステップと、前記コネクタ呼び出し定義情報を参照してコネクタ参照名を取得するステップと、所定の第二のプログラムのインタフェースに整合した形式にデータを変換するステップと、前記変換データをコネクタ参照名に対応した第二のプログラムに渡すステップとを有することを特徴とするプログラム制御方法。

【請求項 2】

第一のプログラムから出力されたデータを入力として第二のプログラムの入力可能な形式に変換して第二のプログラムに出力するアダプタプログラムを有するプログラム制御方法において、

上記アダプタプログラムは、第一のプログラムからデータを含む起動要求を入力するステップと、起動されたアダプタ実行部において、特定の業務で利用するアダプタコンポーネントを呼び出すためのアダプタ呼び出し定義情報を参照してアダプタコンポーネント参照名を取得するステップと、所定のアダプタコンポー

ネットのインタフェースに整合した形式にデータを変換するステップと、前記、変換データをアダプタコンポーネントに渡すステップと、第二のプログラムを呼び出すためのコネクタ呼び出し定義情報を参照してコネクタ参照名を取得するステップと、所定の第二のプログラムのインタフェースに整合した形式にデータを変換するステップと、前記、変換データをコネクタ参照名に対応した第二のプログラムに渡すステップを有することを特徴とするプログラム制御方法。

【請求項 3】

請求項1または請求項2のプログラム制御方法において、

第一のプログラムに複数の業務システムを連携させるワークフロープログラム、第二のプログラムにJ2EE Connectorコンポーネントモデルに基づいたコンポーネントおよびアダプタコンポーネントにEnterprise JavaBeansコンポーネントモデルに基づいたコンポーネントを適用することを特徴とするプログラム制御方法。

【請求項 4】

請求項1または請求項2のプログラム制御方法において、

第一のプログラムに複数の業務システムを連携させるハブアンドスポーク型システム、第二のプログラムにJ2EE Connectorコンポーネントモデルに基づいたコンポーネントおよびアダプタコンポーネントにEnterprise JavaBeansコンポーネントモデルに基づいたコンポーネントを適用することを特徴とするプログラム制御方法。

【請求項 5】

請求項1または請求項2のプログラム制御方法において、

第一のプログラムにWWWからの入力データを処理するWebアプリケーション、第二のプログラムにJ2EE Connectorコンポーネントモデルに基づいたコンポーネントおよびアダプタコンポーネントにEnterprise JavaBeansコンポーネントモデルに基づいたコンポーネントを適用することを特徴とするプログラム制御方法。

【請求項 6】

第一のプログラム実行手段から出力されたデータを入力として第二のプログラム実行手段の入力可能な形式に変換して第二のプログラム実行手段に出力するア

アダプタプログラム実行手段とを備えた計算機システムにおいて、

上記アダプタプログラム実行手段は、第一のプログラム実行手段からデータを含む起動要求を入力する手段と、起動されたアダプタ実行部における複数の業務で共通に利用するアダプタコンポーネント実行手段を呼び出すためのアダプタ呼び出し定義情報を参照してアダプタコンポーネント参照名を取得する手段と、前記アダプタ呼び出し定義情報を参照して業務識別情報を取得する手段と、所定のアダプタコンポーネント実行手段のインタフェースに整合した形式にデータを変換する手段と、業務識別情報と前記変換データをアダプタコンポーネント実行手段に渡す手段と、業務識別情報をもとに第二のプログラム実行手段を起動するためのコネクタ呼び出し定義情報を選択する手段と、前記コネクタ呼び出し定義情報を参照してコネクタ参照名を取得する手段と、所定の第二のプログラム実行手段のインタフェースに整合した形式にデータを変換する手段と、前記変換データをコネクタ参照名に対応した第二のプログラム実行手段に渡す手段とを有することを特徴とする計算機システム。

【請求項 7】

第一のプログラムから出力されたデータを入力として第二のプログラムの入力可能な形式に変換して第二のプログラムに出力するアダプタプログラムを有する制御プログラムを格納した計算機読み取り可能な記録媒体において、

上記アダプタプログラムは、第一のプログラムからデータを含む起動要求を入力するステップと、起動されたアダプタ実行部における複数の業務で共通に利用するアダプタコンポーネントを呼び出すためのアダプタ呼び出し定義情報を参照してアダプタコンポーネント参照名を取得するステップと、前記アダプタ呼び出し定義情報を参照して業務識別情報を取得するステップと、所定のアダプタコンポーネントのインタフェースに整合した形式にデータを変換するステップと、業務識別情報と前記、変換データをアダプタコンポーネントに渡すステップと、業務識別情報をもとに第二のプログラムを呼び出すためのコネクタ呼び出し定義情報を選択するステップと、前記コネクタ呼び出し定義情報を参照してコネクタ参照名を取得するステップと、所定の第二のプログラムのインタフェースに整合した形式にデータを変換するステップと、前記変換データをコネクタ参照名に対応

した第二のプログラムに渡すステップとを有する制御プログラムを格納したことを特徴とする記録媒体。

【請求項 8】

第一のプログラムから出力されたデータを入力として第二のプログラムの入力可能な形式に変換して第二のプログラムに出力するアダプタプログラムを有する制御プログラムにおいて、

上記アダプタプログラムは、第一のプログラムからデータを含む起動要求を入力するステップと、起動されたアダプタ実行部における複数の業務で共通に利用するアダプタコンポーネントを呼び出すためのアダプタ呼び出し定義情報を参照してアダプタコンポーネント参照名を取得するステップと、前記アダプタ呼び出し定義情報を参照して業務識別情報を取得するステップと、所定のアダプタコンポーネントのインタフェースに整合した形式にデータを変換するステップと、業務識別情報と前記、変換データをアダプタコンポーネントに渡すステップと、業務識別情報をもとに第二のプログラムを呼び出すためのコネクタ呼び出し定義情報を選択するステップと、前記コネクタ呼び出し定義情報を参照してコネクタ参照名を取得するステップと、所定の第二のプログラムのインタフェースに整合した形式にデータを変換するステップと、前記変換データをコネクタ参照名に対応した第二のプログラムに渡すステップとを有することを特徴とする制御プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は複数の情報システムを連携させるインテグレーション基盤に、情報システムのインタフェースをラッピングし他のプログラムに対するインタフェースを提供するコネクタコンポーネントを接続するプログラム制御技術に関する。

【0002】

【従来の技術】

近年、ビジネスの変化の速さとボーダレス化、基幹業務システムへの急テンポの変化圧力などから、短期・低コストで新業務サービスが実現でき、かつ、変化

に即応できる柔軟なシステム構造が実現できるシステムが望まれている。そこで既存の流通パッケージや既存情報システム資産を統合的なインテグレーション基盤を用いて組み合わせることにより、企業内の新たな情報システムを迅速かつ確実に実現するエンタープライズアプリケーションインテグレーション（以下、EAI）と呼ばれる分野が米国を中心に立ちあがってきている。インテグレーション基盤は、主に情報システム間の経路制御やおよび情報システム間で流れるデータの受け渡しを行う。情報システム間で流れるデータのことを業務データと呼ぶ。EAIの第一の公知例は、特開2001-16274に記載されている。本公知例では、インテグレーション基盤をハブと呼び、ハブを用いたシステム全体をハブ・アンド・スポークシステムと呼ぶ。

【0003】

EAIの第二の公知例は、日経コンピュータ、1999年9月13日号、第32ページから第ページに記載されている。本公知例では、インテグレーション基盤をワークフローエンジンと呼び、ワークフローエンジンを用いたシステム全体をワークフロー管理システムと呼ぶ。ワークフロー管理システムを用いた情報システム間の連携の一例を図2に示す。このシステムは商品の購入に関わる情報システムである。このシステムは三つの業務ステップから構成される。処理は、まず見積117が行われ、次に発注118が行われ、最後に決済119が行われる。この3つの処理それぞれに対して、既存の情報システムが使われており、それぞれの情報システム間は電子的には接続されていないものとする。以上のように各情報システムを接続して連携させる場合、個別に対処する必要があるが、ワークフロー管理システムを用いることにより、インテグレーション基盤であるワークフローエンジン101が業務の流れおよび業務データの受け渡しを制御するため、迅速にシステムを構築、変更することができる。

【0004】

また近年では、分散オブジェクト技術を利用して構築した情報処理システムである、分散オブジェクトシステムにおいて、サーバプログラムの短期開発を実現するため、プログラムの再利用を可能にするコンポーネントモデルがいくつか登場している。再利用を可能にするため、コンポーネントモデルでは、再利用を可

能にするため、一定の規則に従ってプログラムを作成する。これをコンポーネントと呼ぶ。コンポーネントは、再利用が可能なため、流通部品として販売、購入することが可能である。

【 0 0 0 5 】

コンポーネントモデルに基づくシステムでは、サービスを提供するプログラムがコンポーネントを起動し、サービスを利用するプログラムがコンポーネントを呼び出すことにより望みの処理を実現する。ここで、サービスを提供するプログラムをサービス提供プログラム、サービスを利用するプログラムをサービス利用プログラムと呼ぶ。

【 0 0 0 6 】

コンポーネントモデルに基づくシステムでは、サービス利用プログラムがコンポーネントを呼び出すための識別子を取得する方法が問題になる。ここで、オブジェクトやコンポーネントを呼び出すための識別子を参照情報と呼ぶ。また、サービス利用プログラムが、コンポーネントの参照情報を取得するために利用するオブジェクトをホームと呼ぶ。

またコンポーネントモデルに基づくシステムでは、コンポーネントの実行基盤が必要となる。これをアプリケーションサーバと呼ぶ。

【 0 0 0 7 】

コンポーネントモデルの公知例は、例えば、米サンマイクロシステムズ社発行の「Enterprise JavaBeans Specification, v1.1(1999)」に記載されている（Enterprise JavaBeansは米サンマイクロシステムズ社の登録商標である）。本公知例では、コンポーネントをEnterprise Beanと呼び、ホームをEJBHomeと呼ぶ。またアプリケーションサーバをEJB ContainerおよびEJB Serverと呼ぶ。

またコンポーネントモデルのうち、既存の流通パッケージや既存情報システム資産をラッピングし、他のプログラムから統一的な方法で既存業務プログラムへ接続できるものがある。本発明では、これを特にコネクタコンポーネントモデルと呼ぶ。

【 0 0 0 8 】

コネクタコンポーネントモデルの公知例は、例えば、米サンマイクロシステム

ズ社発行の「J2EE Connector Architecture Specification, Version1.0, Proposed Final Draft#2(2001)」に記載されている。本公知例では、コネクタコンポーネントをConnectorと呼び、ホームをConnectionFactoryと呼ぶ。またアプリケーションサーバをapplication serverと呼ぶ。

【0009】

ここで従来技術においてインテグレーション基盤にコンポーネントを用いて、既存業務プログラムを接続する方法について図1を用いて述べる。

【0010】

一般にインテグレーション基盤101と既存業務プログラム115を統合させる場合、アダプタプログラム126と呼ばれるインテグレーション基盤101と既存業務プログラム115との接続手段を実装する必要がある。アダプタプログラム126はインテグレーション基盤101が規定するアダプタの接続インタフェースに従い、インテグレーション基盤101との接続を開始・保持・終了させる。このため従来、アダプタプログラム126にはインテグレーション基盤101との接続の状態を管理する手段を実装する必要があった。またアダプタプログラム126は、インテグレーション基盤101からアダプタプログラム126へ渡される入力データのデータ構造を既存業務プログラム115の入力データのデータ構造に変換する手段と既存業務プログラム115から出力されるデータのデータ構造をインテグレーション基盤101が要求する出力データのデータ構造に変換する手段を実装する必要があった。さらにアダプタプログラム126は既存業務プログラム115が提供する固有なインターフェースを考慮して既存業務プログラム115との接続を開始・実行・終了させる処理を実装する必要があった。

【0011】

ただし既存業務プログラム115への接続にコネクタコンポーネント113を用いることにより、アダプタプログラム126は既存業務プログラム115に固有なインターフェースを扱う必要がなく、統一的なインターフェースを考慮してコネクタコンポーネント113との接続を開始・実行・終了させる処理を実装することができる。

【0012】

またコネクタコンポーネント113が提供するサービスを利用するサービス利用

プログラムとしてコンポーネント108を用いることにより、アダプタプログラム126の実装の再利用性を向上させることができる。本発明ではこれを特にアダプタコンポーネントと呼ぶ。

【0013】

【発明が解決しようとする課題】

コネクタコンポーネントが既に提供されている状態で、アダプタプログラムを開発する場合、アダプタプログラムを開発する者は、インテグレーション基盤との接続の状態を管理する手段、コネクタコンポーネントが提供しているサービスを利用する手段、アダプタコンポーネントのコンポーネントモデルを実装する手段、各コンポーネント間の複雑な通信インタフェースなどについて熟知している必要がある。またインテグレーション基盤からアダプタへ渡される入力データをコネクタコンポーネントが要求するデータ形式へ変換し、実行結果をインテグレーション基盤に返すときは逆に、インテグレーション基盤が要求するデータ形式へ変換する必要がある。

【0014】

本発明の目的は、複数のコネクタコンポーネントに接続することができるアダプタコンポーネントと、インテグレーション基盤からの実行要求を受け付け、アダプタコンポーネントを起動するアダプタ実行部とを提供することにより、プログラムの保守性を高めることである。

【0015】

【課題を解決するための手段】

本発明は、コネクタコンポーネントに接続することができるアダプタコンポーネントと、インテグレーション基盤からの実行要求を受け付け、アダプタコンポーネントを起動するアダプタ実行部の二つをアダプタプログラムとして提供することにより、インテグレーション基盤から出力されたデータを入力として、コネクタコンポーネントの入力可能な形式に変換してコネクタコンポーネントに出力するプログラムの制御方法の特徴とする。

【0016】

上記アダプタプログラムは、インテグレーション基盤からデータを含む起動要求を入力するステップと、起動されたアダプタ実行部において、アダプタコンポーネントを呼び出すためのアダプタ呼び出し定義情報を参照してアダプタコンポーネント参照名を取得するステップと、所定のアダプタコンポーネントのインタフェースに整合した形式にデータを変換するステップと、前記、変換データをアダプタコンポーネントに渡すステップと、コネクタコンポーネントを呼び出すためのコネクタ呼び出し定義情報を参照してコネクタ参照名を取得するステップと、所定のコネクタコンポーネントのインタフェースに整合した形式にデータを変換するステップと、前記、変換データをコネクタ参照名に対応したコネクタコンポーネントに渡すステップを有することを特徴とする。

【 0 0 1 7 】

【発明の実施の形態】

本発明の実施例について図面を用いて説明する。

第一の実施例について図1から図8を用いて説明する。第一の実施例は、コネクタコンポーネントが既に提供されている状態でアダプタを開発する場合における以下の課題を改善するものである。

【 0 0 1 8 】

まずコネクタコンポーネントが既に提供されている状態で、アダプタプログラムを開発する場合、アダプタプログラムを開発する者は、インテグレーション基盤との接続の状態を管理する手段、コネクタコンポーネントが提供しているサービスを利用する手段、アダプタコンポーネントのコンポーネントモデルを実装する手段、各コンポーネント間の複雑な通信インタフェースなどについて熟知している必要がある。またインテグレーション基盤からアダプタへ渡される入力データをコネクタコンポーネントが要求するデータ形式へ変換し、実行結果をインテグレーション基盤に返すときは逆に、インテグレーション基盤が要求するデータ形式へ変換する必要がある。

【 0 0 1 9 】

またアダプタコンポーネントはインテグレーション基盤が定義する各業務ごとに実装する必要がある。複数の業務で同一のコネクタコンポーネントを用いる場

合でも、インテグレーション基盤とアダプタとの間で授受される入出力データ項目が異なる場合や、コネクタコンポーネントが提供するメソッド群のうち、インテグレーション基盤からの実行要求に応えるメソッドが異なる場合、各業務ごとに実装する必要がある。さらにインテグレーション基盤が定義する業務やコネクタコンポーネントが提供するサービスが変更された場合、アダプタコンポーネントの実装を変更しなければならない。

【0020】

本実施例の目的は、複数のコネクタコンポーネントに接続することができるアダプタコンポーネントと、インテグレーション基盤からの実行要求を受け付け、アダプタコンポーネントを起動するアダプタ実行部の二つを提供することにより、システム開発の工数を削減し、プログラムの保守性を高めることである。

【0021】

次に本実施例のシステムの概要を述べる。

図1はインテグレーション基盤101と各業務を実行する既存業務システム115を中心としたシステム群を示している。例えば、見積117を実行する既存業務システム115は見積業務システム124であり、見積業務システム124のインタフェースをラッピングしたコネクタコンポーネント113は見積業務用コネクタコンポーネント120である。同様に、発注118を実行する既存業務システム115は発注業務システム125であり、発注業務システム125のインタフェースをラッピングしたコネクタコンポーネント113は発注業務用コネクタコンポーネント121である。決済119を実行する既存業務システム115は決済業務システム126であり、決済業務システム126のインタフェースをラッピングしたコネクタコンポーネント113は決済業務用コネクタコンポーネント122である。

【0022】

本実施例では、インテグレーション基盤101が提供するインテグレーション基盤メタ定義情報102を読み込み、コネクタコンポーネント113が提供するコネクタメタデータ定義情報114を読み込み、コネクタコンポーネント113が提供するメソッド群のうち、インテグレーション基盤101からの実行要求に応えるメソッドをユーザに選択させることにより、アダプタ呼び出し定義情報107およびコネクタ

呼び出し定義情報112をアダプタプログラムが実行時に参照することができるファイルまたはアダプタプログラムの一部として生成する。またインテグレーション基盤101が定義する入出力データ項目とコネクタコンポーネント113が定義する入出力データ項目についてユーザにデータ項目のマッピングを行わせることにより、データマッピング定義情報をコネクタ呼び出し定義情報112の一部として生成する。

【0023】

実行時には、アダプタ実行部103がインテグレーション基盤101の特定の業務から起動され、入力データが渡される。次に起動されたアダプタ実行部103は、アダプタ呼び出し定義情報107から複数の業務で共通に利用するアダプタコンポーネント108の参照名および業務の識別情報、入出力データのメタデータ定義情報を取得する。それをもとにアダプタ実行部103は、インテグレーション基盤101から渡された入力データをアダプタコンポーネント108のインタフェースに整合した形式にデータを変換し、アダプタコンポーネント108の参照名を用いてアダプタコンポーネント108を呼び出し、前記、業務識別名、変換データを渡す。次にアダプタコンポーネント108は、業務識別名をことに業務を実行するコネクタコンポーネント113を呼び出すためのコネクタ呼び出し定義情報112を選択し、コネクタ呼び出し定義情報112からコネクタコンポーネント113の参照名および入出力データのメタデータ定義情報を取得する。それをもとにアダプタコンポーネント108は、入力データをコネクタコンポーネント113のインタフェースに整合した形式にデータを変換し、コネクタコンポーネント113の参照名を用いてコネクタコンポーネント113を呼び出し、前記、変換データを渡す。

【0024】

次に、本システムの全体構成の概要を述べる。

図2はアダプタ呼び出し定義情報107およびコネクタ呼び出し定義情報112を生成し、インテグレーション基盤101を中心とするシステムを実行するインテグレーション基盤サーバ201の構成図を示す。インテグレーション基盤サーバ201は、主記憶202、二次記憶装置203、CPU 204、ディスプレイ205、キーボード206、マウス207、ネットワークインタフェース208と、これらを相互接続するバス209から

なる。そして、インテグレーション基盤サーバ201は、ネットワークインタフェース208を介してネットワーク210に接続される。

【 0 0 2 5 】

また主記憶202の中に、オペレーティングシステム（以下、OS）211、本発明の課題を改善するアダプタ定義用プログラム群220、インテグレーション基盤101、コネクタコンポーネント113、既存業務プログラム115、アダプタ実行部103、アダプタコンポーネント108が記憶されている。さらに二次記憶装置203の中に、インテグレーション基盤メタデータ定義情報102、コネクタメタデータ定義情報112、アダプタ呼び出し定義情報107、コネクタ呼び出し定義情報112が記憶されている。

【 0 0 2 6 】

アダプタ定義用プログラム群220は、インテグレーション基盤メタデータ定義読込手続き212、コネクタメタデータ定義読込手続き213、定義ツール214、コネクタ呼び出し定義情報生成手続き218、アダプタ呼び出し定義情報生成手続き219からなる。インテグレーション基盤メタデータ定義読込手続き212はインテグレーション基盤メタデータ定義情報102の読み込みを行う。コネクタメタデータ定義読込手続き213はコネクタメタデータ定義情報111の読み込みを行う。定義ツール214は業務選択ツール215、業務処理メソッド選択ツール216、データマッピング定義ツール217からなる。業務選択ツール215はインテグレーション基盤メタデータ定義情報102に定義された業務群のうち、ユーザに自動生成するアダプタコンポーネント106が対応する業務をユーザに選択させるツールである。業務処理メソッド選択ツール216は、コネクタメタデータ定義情報111に定義されたメソッド群のうち、インテグレーション基盤からの実行要求に応えるメソッドをユーザに選択させるツールである。データマッピング定義ツール217は、業務選択ツール215で選択した業務の入出力データと、業務処理メソッド選択ツール216で選択した業務処理メソッドの入出力データとの間でのマッピングをユーザに行わせるツールである。コネクタ呼び出し生成手続き218は、コネクタメタデータ定義情報112、業務選択ツール215で選択した業務、業務処理メソッド選択ツール216で選択した業務処理メソッド、データマッピング定義情報を利用して、アダプタコ

ンポーネント108がコネクタコンポーネント113を呼び出すためのコネクタ呼び出し手続き112を生成する。アダプタ呼び出し生成手続き219は、インテグレーション基盤メタデータ定義情報101、業務選択ツール215で選択した業務を利用して、アダプタ実行部103がアダプタコンポーネント108を呼び出すためのアダプタ呼び出し定義情報107を生成する。

【 0 0 2 7 】

次に各定義や定義ツールについて詳細に説明する。

図3はインテグレーション基盤メタデータ定義情報102の詳細を示す。インテグレーション基盤定義情報102は、アダプタ実行部102など外部からインテグレーション基盤101を参照するための基盤参照名301、インテグレーション基盤101上で定義された各業務の内容を示す業務302からなる。業務302は業務の名前を示す業務名303、業務データの内容を示す業務データ304からなる。業務データ304は、業務の入出力データを表す入力データ305および出力データ306からなり、入力データ305および出力データ306はデータ項目のデータ名称を示す項目307、データ項目の型を示す型名308、データ項目の出現順序を示す順序309からなる。

【 0 0 2 8 】

図4はコネクタメタデータ定義情報114の詳細を示す。コネクタメタデータ定義情報114は、サービス利用プログラムが、コネクタコンポーネント113の参照情報を取得するためのホーム名称401、コネクタコンポーネント113上で定義された各メソッドの内容を示すメソッド402からなる。メソッド402はメソッドの名前を示すメソッド名403、メソッドの入出力データの内容を示す入出力データ404からなる。入出力データ404は、入力データ405および出力データ406からなり、入力データ405および出力データ406はデータ項目のデータ名称を示す項目407、データ項目の型を示す型名408、データ項目の出現順序を示す順序409からなる。

【 0 0 2 9 】

図5は定義ツール214の例を示す。定義ツール214は業務選択ツール215、業務処理メソッド選択ツール216、データマッピング定義ツール217からなる。業務選択ツール215は、インテグレーション基盤メタデータ定義読み込み手続き212が読み込んだインテグレーション基盤メタデータ定義情報102の各業務の名前を示す業

務名501および業務の入力データのデータ項目名を示すデータ項目502, データ項目の型を示す型名503, 出力データのデータ項目名を示すデータ項目504, データ項目の型を示す型名505など業務の内容を表示し, ユーザに業務を選択させる。業務処理メソッド選択ツール216は, コネクタメタデータ定義読み込み手続き213が読み込んだコネクタメタデータ定義情報112の各メソッドの名前を示すメソッド名506およびメソッドの入力データのデータ項目名を示すデータ項目507, データ項目の型を示す型名508, 出力データのデータ項目名を示すデータ項目509, データ項目の型を示す型名510など業務の内容を表示し, ユーザに業務処理メソッドを選択させる。データマッピング定義ツール217は業務選択ツール515によって選択された業務の入出力データ項目および業務処理メソッド選択ツール516によって選択された業務処理メソッドの入出力データ項目の一覧を表示に, ユーザにデータの対応関係を指定させる。

【 0 0 3 0 】

図6はアダプタ呼び出し定義情報107およびコネクタ呼び出し定義情報112を含む呼び出し定義情報116の詳細を示す。アダプタ呼び出し定義情報107は, インテグレーション基盤101から処理要求を受け取るあるいは処理結果を返すために利用するインテグレーション基盤101の参照名を示す基盤参照名601, アダプタコンポーネント108を呼び出すために利用するアダプタコンポーネント108の参照名を示すアダプタ参照名602, 業務選択ツール215によって選択された業務の名前を示す業務名603, 業務の入出力データの内容を示す業務データ604からなる。基盤参照名601は, インテグレーション基盤メタデータ定義情報102から読み込んだ基盤参照名301の値を利用する。また業務データ604は, 業務の入出力データを表す入力データ605および出力データ606からなり, 入力データ605および出力データ606はデータ項目のデータ名称を示す項目607, データ項目の型を示す型名608, データ項目の出現順序を示す順序609からなる。

【 0 0 3 1 】

次にコネクタ呼び出し定義情報112は, 業務選択ツール215によって選択された業務の名前を示す業務識別名610, コネクタコンポーネント113を呼び出すために利用するコネクタコンポーネント113の参照名を示すコネクタ参照名611, 業務メ

ソッド選択ツール216によって選択された業務処理メソッドの名前を示すメソッド名612, 業務処理メソッドの入出力データの内容を示す入出力データ613からなる。コネクタ参照名611は, コネクタメタデータ定義情報114から読み込んだホーム名称401の値を利用する。入出力データ613は, 入力データ614および出力データ615からなり入力データ614および出力データ615はデータ項目のデータ名称を示す項目616, データ項目の型を示す型名617, データ項目の出現順序を示す順序618, データマッピング定義ツール217で選択した業務メソッドの入出力データと対応するインテグレーション基盤101で定義された業務の入出力データ項目619からなる。

【0032】

次にアダプタ呼び出し定義情報107およびコネクタ呼び出し定義情報112を生成する処理の流れを図7を用いて説明する。

【0033】

(ステップ701) インテグレーション基盤メタデータ定義読込手続き212は, インテグレーション基盤101からインテグレーション基盤メタデータ定義情報102を読み込み, インテグレーション基盤で定義された各業務と各業務を処理する情報システムとの間で授受される入出力データ項目のメタ情報を読み込む。図3のインテグレーション基盤メタ定義情報102を例にメタ情報を説明すると, アダプタプログラム126がインテグレーション基盤101に結果を返すための基盤参照名301として「ワークフロー」という値を読み込む。次にインテグレーション基盤101上で定義された業務302の業務名303として「見積」や「発注」という値を読み込む。次に各業務ごとに入出力データ304として「見積番号」や「部品番号」, 「発注金額」という値を読み込む。入力データ305および出力データ306は, 項目307, 型名308, 順序309という項目からなる。例えば項目307の一つである「見積番号」の型名308は「文字列」であり, 入力データ305の中における順序309は「1」である。

【0034】

(ステップ702) ステップ701で読み込んだインテグレーション基盤メタデータ定義情報102をもとに, 業務選択ツール215を用いて, インテグレーション基盤メ

アダプタ定義情報102に定義された業務群のうち、ユーザに自動生成するアダプタコンポーネント108が対応する業務をユーザに選択させる。図5の業務選択ツール215を例に説明すると、画面上にインテグレーション基盤メタデータ定義情報102から読み込んだメタ情報をもとに業務名501と業務名501に対応する入力データのデータ名称502や型名503、出力データのデータ名称504や型名505などを表示させ、ユーザに業務名501を選択させる。

(ステップ703) ステップ701で読み込んだインテグレーション基盤メタデータ定義情報102およびステップ702で選択した業務名をもとにアダプタ呼び出し定義情報107を生成する。図6のアダプタ呼び出し定義情報107を例に説明すると、基盤参照名601はステップ701で読み込んだインテグレーション基盤メタデータ定義情報102の基盤参照名301「ワークフロー」とする。アダプタ参照名602は固定的に「共通アダプタ」という名前にしておき、実行時にアダプタコンポーネントを「共通アダプタ」という名前で参照できるようにする。業務名603、入出力データ604等は、ステップ702でユーザに選択させた業務の値とする。例えばステップ702で「発注」を選択した場合、業務名603が「発注」となり、入力データ605の項目607は「見積番号」，「部品番号」，「発注金額」となる。

【0035】

(ステップ704) コネクタメタデータ定義読込手続き213は、コネクタコンポーネント113からコネクタメタデータ定義情報114を読み込み、コネクタコンポーネント113が提供するコネクタのホーム名称といったインタフェース情報や入出力データ項目のメタ情報を取得する。図4のコネクタメタ定義情報114を例にメタ情報を説明すると、アダプタプログラム126がコネクタコンポーネント113を参照するためのホーム名称401として「受発注システム」という値を読み込む。次にインテグレーション基盤101上で定義されたメソッド402のメソッド名403として「見積登録」や「発注登録」という値を読み込む。次に各メソッドごとに入出力データ403として「番号」や「金額」，「部品名」という値を読み込む。入力データ405および出力データ406は、項目407、型名408、順序409という項目からなる。例えば項目407の一つである「番号」の型名408は「文字列」であり、入力データ405の中における順序409は「1」である。

【 0 0 3 6 】

(ステップ705) ステップ704で読み込んだコネクタメタデータ定義情報114をもとに、業務処理メソッド選択ツール216を用いて、ユーザにコネクタコンポーネント113が提供するメソッド群のうち、インテグレーション基盤からの実行要求に応える業務処理メソッドをユーザに選択させる。図5の業務処理メソッド選択ツール216を例に説明すると、画面上にコネクタメタ定義情報114から読み込んだメタ情報をもとにメソッド名504とメソッド名504に対応する入力データのデータ名称507や型名508、出力データのデータ名称509や型名510などを表示させ、ユーザにメソッド名504を選択させる。

(ステップ706) ステップ702で選択した該当業務に入力データが定義されているかどうかチェックする。入力データがある場合はステップ707へ、入力データがない場合はステップ708へ進む。

【 0 0 3 7 】

(ステップ707) データマッピング定義ツール217を用いてユーザにステップ702で選択した業務の入力データと、ステップ705で選択した業務処理メソッドの入力データとの間でマッピングを行わせる。図5のデータマッピング定義ツール217を例に説明すると、ステップ702で選択した業務名501「発注」の入力データが業務データ入力511として表示される。次にステップ705で選択したメソッド名506「発注登録」の入力データがコネクタデータ入力512として表示される。そして業務データ入力511とコネクタデータ入力512の各項目間でデータのマッピングをユーザに行わせる。例えば「見積番号」を「番号」に、「部品番号」を「部品」に、「発注金額」を「金額」に対応させる。

【 0 0 3 8 】

(ステップ708) ステップ702で選択した該当業務に出力データが定義されているかどうかチェックする。出力データがある場合はステップ709へ、出力データがない場合はステップ710へ進む。

【 0 0 3 9 】

(ステップ709) データマッピング定義ツール217を用いてユーザにステップ702で選択した業務の出力データと、ステップ705で選択した業務処理メソッドの出

カデータとの間でマッピングを行わせる。図5のデータマッピング定義ツール217を例に説明すると、ステップ705で選択したメソッド名506「発注登録」の出力データがコネクタデータ出力513として表示される。次にステップ702で選択した業務名501「発注」の出力データが業務データ出力514として表示される。そしてコネクタデータ出力513と業務データ出力514の各項目間でデータのマッピングをユーザに行わせる。例えば「番号」を「発注番号」に、「金額」を「発注金額」に、「部品名」を「部品番号」に対応させる。

【0040】

(ステップ710) ステップ704で読み込んだコネクタメタデータ定義情報114, ステップ705で選択した業務処理メソッド, ステップ706およびステップ708でユーザが行ったデータマッピングをもとにコネクタ呼び出し定義情報112を生成する。図6のコネクタ呼び出し定義情報112を例に説明すると、業務参照名610はステップ702で選択した業務名501とする。コネクタ参照名611はステップ704で読み込んだコネクタメタデータ定義情報112のホーム参照名401「受発注システム」とする。メソッド名612, 入出力データ613等は, ステップ705でユーザに選択させたメソッドの値とする。例えばステップ705で「発注登録」を選択した場合, メソッド名612が「発注登録」となり, 入力データ614の項目616は「番号」, 「金額」, 「部品名」となる。また対応業務データ名619はステップ706およびステップ708でユーザが行ったデータマッピングによるインテグレーション基盤101で定義された対応する入出力データの項目名とする。例えば, 「番号」には「見積番号」, 「金額」には「発注金額」, 「部品名」には「部品番号」が対応する。

【0041】

次にインテグレーション基盤を中心とした実行時の流れを図8を用いて説明する。

【0042】

(ステップ801) インテグレーション基盤101の特定の業務は, アダプタ実行部103を起動し, 入力データ821を渡す。例えば, 図1のインテグレーション基盤101で定義された業務である発注118は, 発注業務を実行するアダプタ実行部103を起動する。そして入力データとして図3のインテグレーション基盤メタ定義情報102

で示される入力データ305「見積番号」，「部品番号」，「発注金額」を渡す。
また入力データ821のデータ形式はインテグレーション基盤101が定めたデータ形式とする。

【0043】

（ステップ802）アダプタ実行部103は，アダプタ呼び出し定義情報107から複数の業務で共通に利用するアダプタコンポーネント108の参照名および業務の識別情報，入力データのメタデータ定義情報を取得する。それをもとにアダプタ実行部103のデータ形式変換部105は，インテグレーション基盤101から渡された入力データ821をアダプタコンポーネント108のインタフェースに整合した形式にデータを変換する。例えば，図6のアダプタ呼び出し定義情報107からアダプタ参照名602「共通アダプタ」，業務の識別情報となる業務名603「発注」，入力データのメタデータ定義情報となる入力データ605「見積番号」，「部品番号」，「発注番号」の値を取得する。それをもとにデータを変換する。

【0044】

（ステップ803）アダプタ実行部103のアダプタ接続部106は，アダプタコンポーネント108の参照名を用いてアダプタコンポーネント108を呼び出し，前記，業務識別名，変換した入力データ822を渡す。例えば，ステップ802で取得したアダプタ参照名602「共通アダプタ」を用いてアダプタコンポーネント108を呼び出し，業務識別名として業務名603「発注」，入力データ822として変換した入力データ605「見積番号」，「部品番号」，「発注番号」を渡す。

【0045】

（ステップ804）アダプタコンポーネント108は，業務識別名をことに業務を実行するコネクタコンポーネント113を呼び出すためのコネクタ呼び出し定義情報112を選択し，コネクタ呼び出し定義情報112からコネクタコンポーネント113の参照名およびメソッド名，入力データのメタデータ定義情報およびインテグレーション基盤101から渡される入力データとの対応関係を取得する。それをもとにアダプタコンポーネント108のデータマッピング部109は入力データを並び替え，データ形式変換部110は入力データ822をコネクタコンポーネント113のインタフェースに整合した形式にデータを変換する。例えば，ステップ803で渡された業務

識別名「発注」と、図6のコネクタ呼び出し定義情報112の業務識別名611を比較し、同じ名前のコネクタ呼び出し定義情報112を選択する。そしてコネクタ参照名611「受発注システム」、メソッド名612「発注登録」、入力データのメタデータ定義情報となる入力データ614「番号」、「金額」、「部品名」、さらに各項目616に対応する対応業務データ名619などの値を取得する。それをもとに入力データ822をコネクタコンポーネント113で定義された順序に並び替えて「番号」、「金額」、「部品名」とし、データを変換する。

【 0 0 4 6 】

(ステップ805) アダプタコンポーネント108コネクタ接続部は、コネクタコンポーネント113の参照名を用いてコネクタコンポーネント113を呼び出し、前記、変換した入力データ823を渡す。例えば、ステップ804で取得したコネクタ参照名611「受発注システム」を用いて図1の発注業務用コネクタコンポーネント121を呼び出し、メソッドとしてメソッド名612「発注登録」、入力データ823として変換した入力データ入力データ614「番号」、「金額」、「部品名」を渡す。

【 0 0 4 7 】

(ステップ806) コネクタコンポーネント113は、入力データ823を既存業務システム115のインタフェースに整合した形式に入力データ823を変換する。例えば、図1の発注業務用コネクタコンポーネント121は、発注業務システム124のインタフェースに整合した形式に入力データ823「番号」、「金額」、「部品名」を変換する。

(ステップ807) コネクタコンポーネント113は、既存業務システム115を呼び出し、前記、変換した入力データ824を渡す。例えば、図1の発注業務用コネクタコンポーネント121は、発注業務システム124を呼び出し、変換した入力データ824「番号」、「金額」、「部品名」を渡す。

【 0 0 4 8 】

(ステップ808) 既存業務システム115は、業務処理を実行する。例えば、図1の発注業務システム124は、入力データ824「番号」、「金額」、「部品名」を用いて業務処理を実行する。

【 0 0 4 9 】

(ステップ809) 既存業務システム115は、業務処理を実行した結果となる出力データ825をコネクタコンポーネント113に渡す。例えば、図1の発注業務システム124は、発注業務用コネクタコンポーネント121に出力データ825「番号」, 「金額」, 「部品名」を渡す。

【0050】

(ステップ810) コネクタコンポーネント113は、出力データ825をコネクタコンポーネント113のインタフェースに整合した形式にデータを変換する。例えば、図1の発注業務用コネクタコンポーネント121は、発注業務システム124から渡された出力データ825「番号」, 「金額」, 「部品名」を変換する。

【0051】

(ステップ811) コネクタコンポーネント113は、前記、変換した入力データ826をアダプタコンポーネント108に渡す。例えば、図1の発注業務用コネクタコンポーネント121は、アダプタコンポーネント108に変換した出力データ826「番号」, 「金額」, 「部品名」を渡す。

【0052】

(ステップ812) アダプタコンポーネント108は、コネクタ呼び出し定義情報112を用いて出力データ826をアダプタコンポーネント108のインタフェースに整合した形式にデータを変換する。例えば、図6のコネクタ呼び出し定義情報112を利用して出力データのメタデータ定義情報となる出力データ615「番号」, 「金額」, 「部品名」, さらに各項目616に対応する対応業務データ名619などの値を取得する。それをもとに出力データ825をインテグレーション基盤111で定義された順序に並び替えて「発注番号」, 「部品番号」, 「発注番号」とし、データを変換する。

【0053】

(ステップ813) アダプタコンポーネント108は、前記、変換した出力データ827をアダプタ実行部103に渡す。例えば、アダプタコンポーネント108は、アダプタ実行部103に変換した出力データ827「発注番号」, 「部品番号」, 「発注番号」を渡す。

(ステップ814) アダプタ実行部103は、出力データ827をインテグレーション基

盤101のインタフェースに整合した形式にデータを変換する。例えば、図6のアダプタ呼び出し定義情報107から出力データのメタデータ定義情報となる出力データ606「発注番号」，「部品番号」，「発注番号」の値を取得する。それをもとにデータを変換する。

（ステップ815）アダプタ実行部103のインテグレーション基盤接続部104は、前記、変換した入力データ828をインテグレーション基盤101に渡す。例えば、アダプタ実行部103は、アダプタ実行部103の基盤参照名601「ワークフロー」を用いてインテグレーション基盤101を呼び出し、変換した出力データ828「発注番号」，「部品番号」，「発注番号」を渡す。

【 0 0 5 4 】

なお第一の実施例において、アダプタ実行部103、アダプタコンポーネント108から構成されるアダプタプログラム126はインテグレーション基盤101からデータを含む起動要求を入力することに限定するものではなく、本実施例はWWWからの入力データを処理するWebアプリケーションにも適用可能である。この場合、インテグレーション基盤の各業務がWWWの各画面からの実行要求に相当し、Webアプリケーションは画面の入出力データについてインテグレーション基盤メタデータ定義情報102相当のメタデータ定義情報を持っている必要がある。

【 0 0 5 5 】

第一の実施例では、インテグレーション基盤から複数の業務の実行要求を受け付け、複数のコネクタコンポーネントに接続することができるアダプタコンポーネントと、インテグレーション基盤からの実行要求を受け付け、アダプタコンポーネントを起動するアダプタ実行部の二つを提供することにより、アダプタプログラムを開発する者が複数のコンポーネントモデルについて熟知する必要がなくなる。さらにシステム開発の工数を削減し、プログラムの保守性を高めることができる。

【 0 0 5 6 】

次に第二の実施例について図9から図10を用いて説明する。第一の実施例では、アダプタ実行部からアダプタコンポーネントに業務の識別名を渡し、アダプタコンポーネントが業務の識別名に応じて特定のコネクタ呼び出し定義情報を選択

していた。第一の実施例は典型的には図9で示す構成で運用する。まずインテグレーション基盤101の各業務から起動されるアダプタ実行部をそれぞれ別々のオブジェクトとして運用する。例えば、見積117を実行するのがアダプタ実行部901，発注118を実行するのがアダプタ実行部902などである。次にアダプタコンポーネントは一つまたは複数のアプリケーションサーバの中で複数の業務の処理を受け付ける共通のオブジェクトとして運用される。しかしインテグレーション基盤を中心とする一連の情報システムに対して、高い処理性能が求められる場合があり、アダプタコンポーネントが複数の業務の処理を受け付ける共通のオブジェクトとして運用することが問題になる場合がある。

【 0 0 5 7 】

本実施例では、業務ごとに別々のアプリケーションサーバを用意し、その中で特定の業務を受け付けるようにアダプタコンポーネントのオブジェクトを運用するものである。本実施例は典型的には図10で示すように運用する。インテグレーション基盤101の各業務に対応したアダプタコンポーネントを別々のアプリケーションサーバで運用する。例えば見積117を実行するのがアプリケーションサーバ1001中のアダプタコンポーネント1005などであり、さらにアダプタコンポーネント1105は見積業務用コンポーネント120のみを呼び出す。

【 0 0 5 8 】

本実施例のシステム構成、各定義や定義ツール、処理の流れは第一の実施例と同様である。ただし、アダプタコンポーネント108は特定のコネクタコンポーネント113を実行するように運用されるため、業務の識別名を用いてコネクタ呼び出し定義情報112を選択する必要はない。アダプタ実行部103が特定の業務を実行するアダプタコンポーネント108を起動すればよい。そのため、コネクタ呼び出し定義情報112に業務識別名610が含まれない場合がある。

【 0 0 5 9 】

なお第二の実施例において、アダプタ実行部103、アダプタコンポーネント108から構成されるアダプタプログラム126はインテグレーション基盤101からデータを含む起動要求を入力することに限定するものではなく、本実施例はWWWからの入力データを処理するWebアプリケーションにも適用可能である。この場合、イ

ンテグレーション基盤の各業務がWWWの各画面からの実行要求に相当し、Webアプリケーションは画面の入出力データについてインテグレーション基盤メタデータ定義情報102相当のメタデータ定義情報を持っている必要がある。

【0060】

第二の実施例では、特定の業務の実行要求を受け付け、特定のコネクタコンポーネントに接続するようにアダプタコンポーネントを運用することで、より高い処理性能が期待できる。

【0061】

【発明の効果】

本発明によれば、複数のコネクタコンポーネントに接続することができるアダプタコンポーネントと、インテグレーション基盤からの実行要求を受け付け、アダプタコンポーネントを起動するアダプタ実行部とを提供することにより、プログラムの保守性を高めることができる。

【図面の簡単な説明】

【図1】 第一の実施例により生成した実行環境の概要を示す図

【図2】 第一の実施例における開発環境、実行環境の構成図

【図3】 アダプタ呼び出し定義情報の内容を示す図

【図4】 コネクタ呼び出し定義情報の内容を示す図

【図5】 定義ツールの一例を示す図

【図6】 第一の実施例における呼び出し定義情報の内容を示す図

【図7】 第一の実施例における開発環境の処理の流れを示す図

【図8】 第一の実施例における実行環境の処理の流れを示す図

【図9】 第一の実施例により生成したプログラムを運用する際の概要を示す図

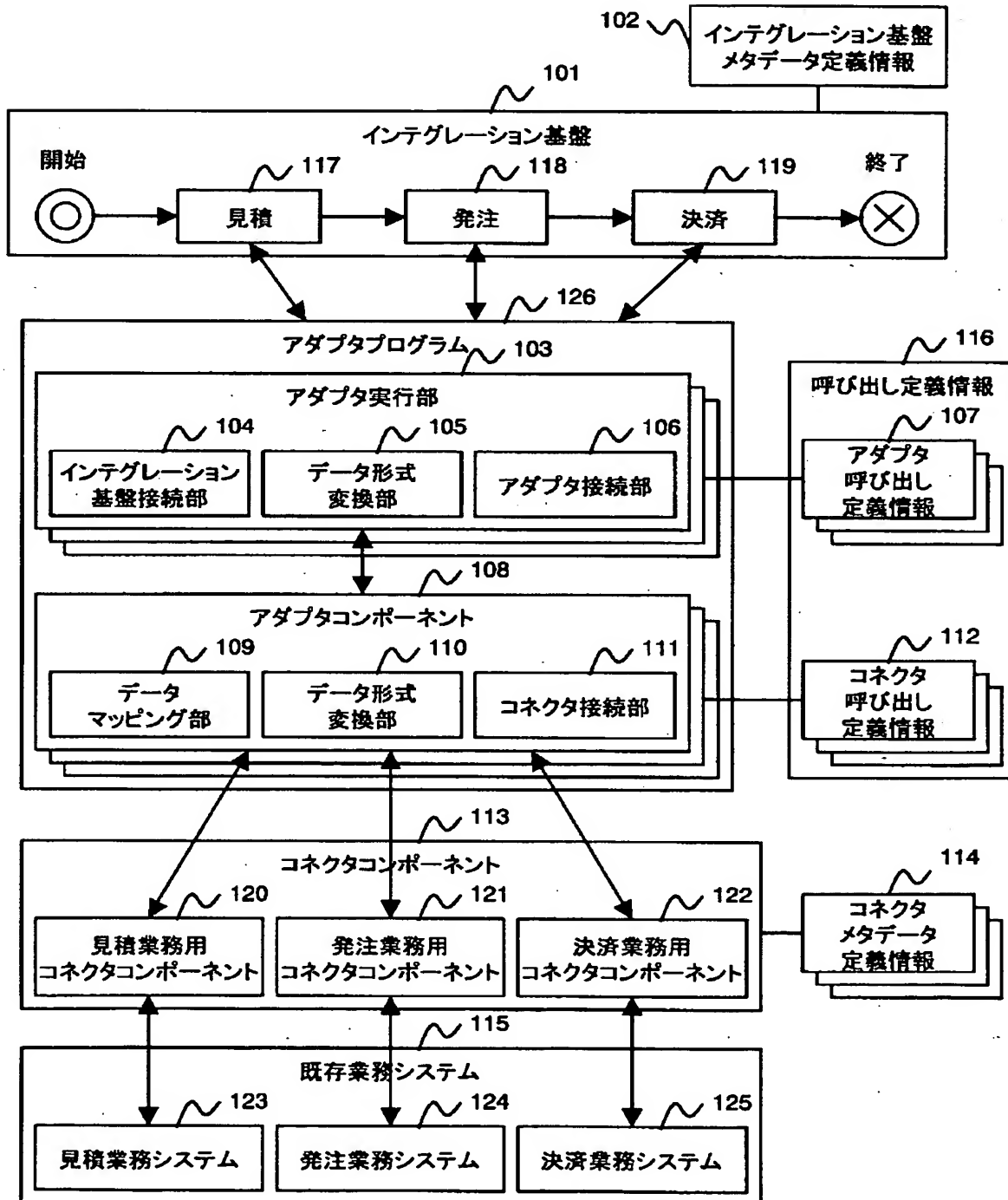
【図10】 第二の実施例により生成したプログラムを運用する際の概要を示す図

【符号の説明】

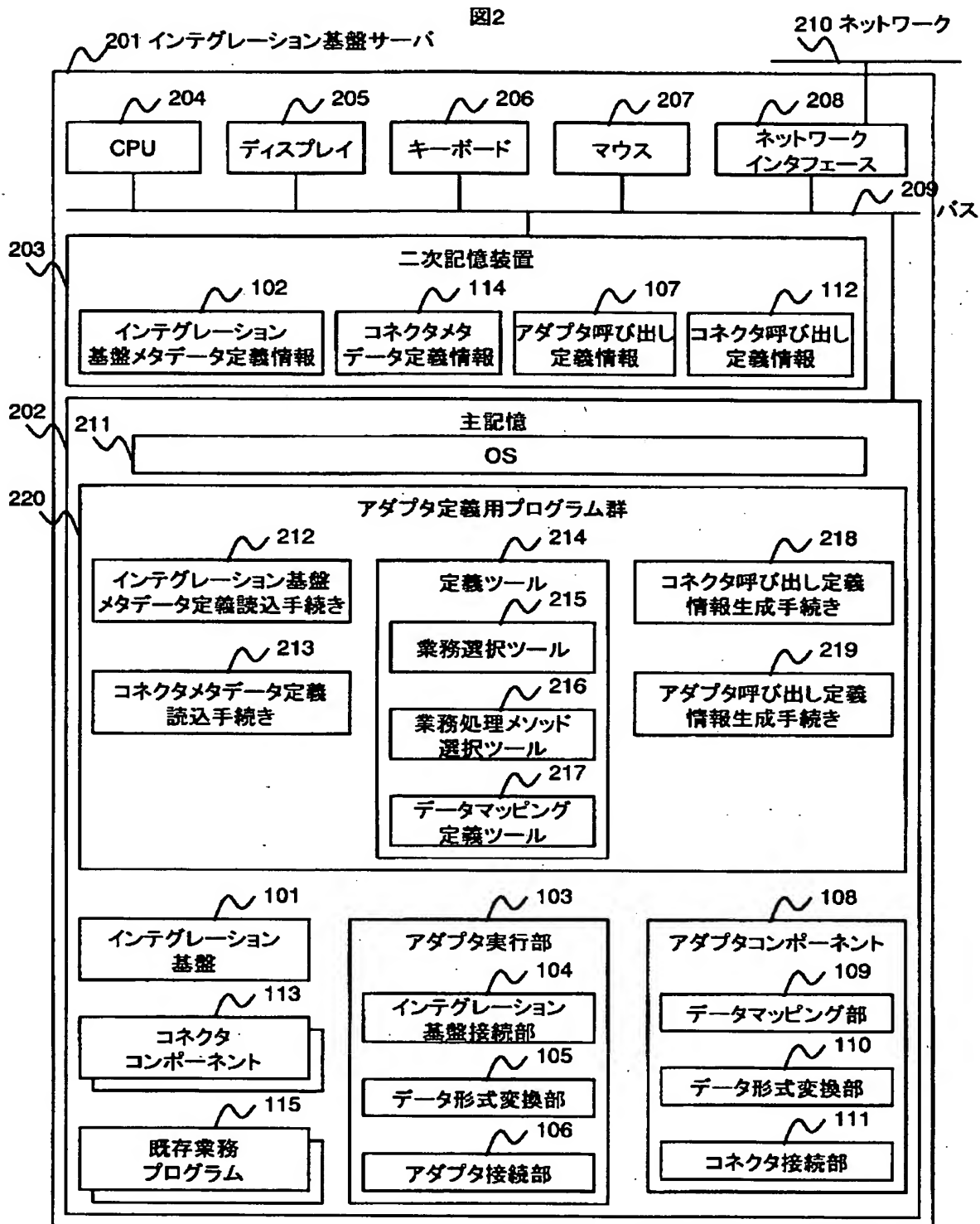
【書類名】 図面

【図1】

図1

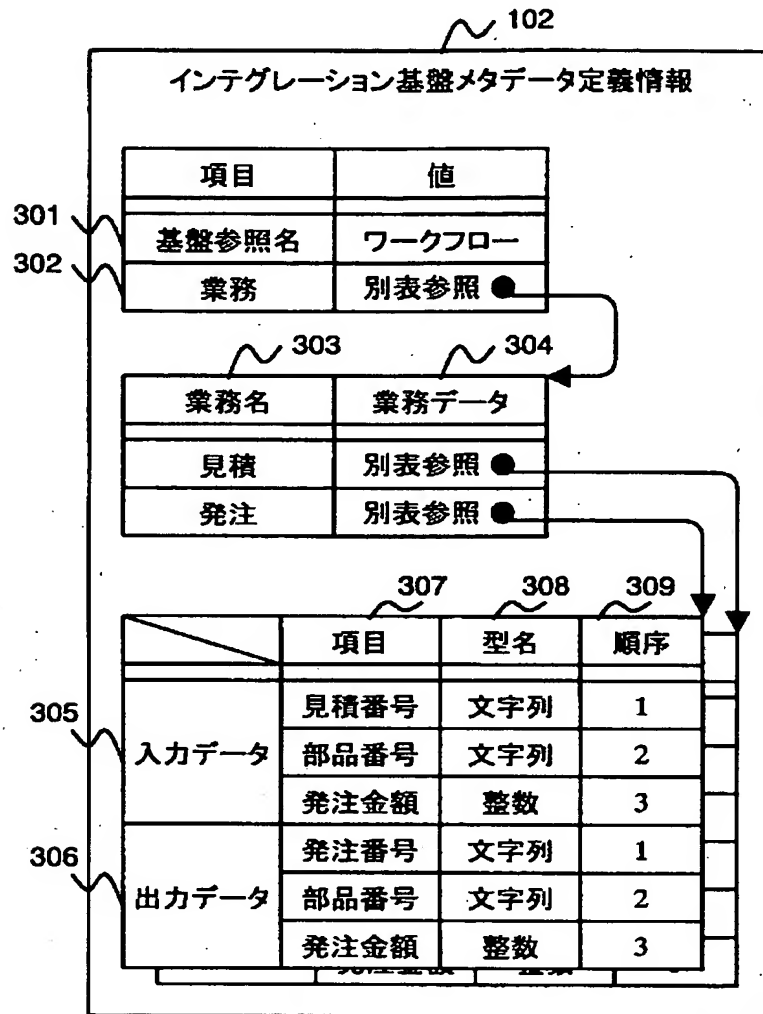


【図 2】



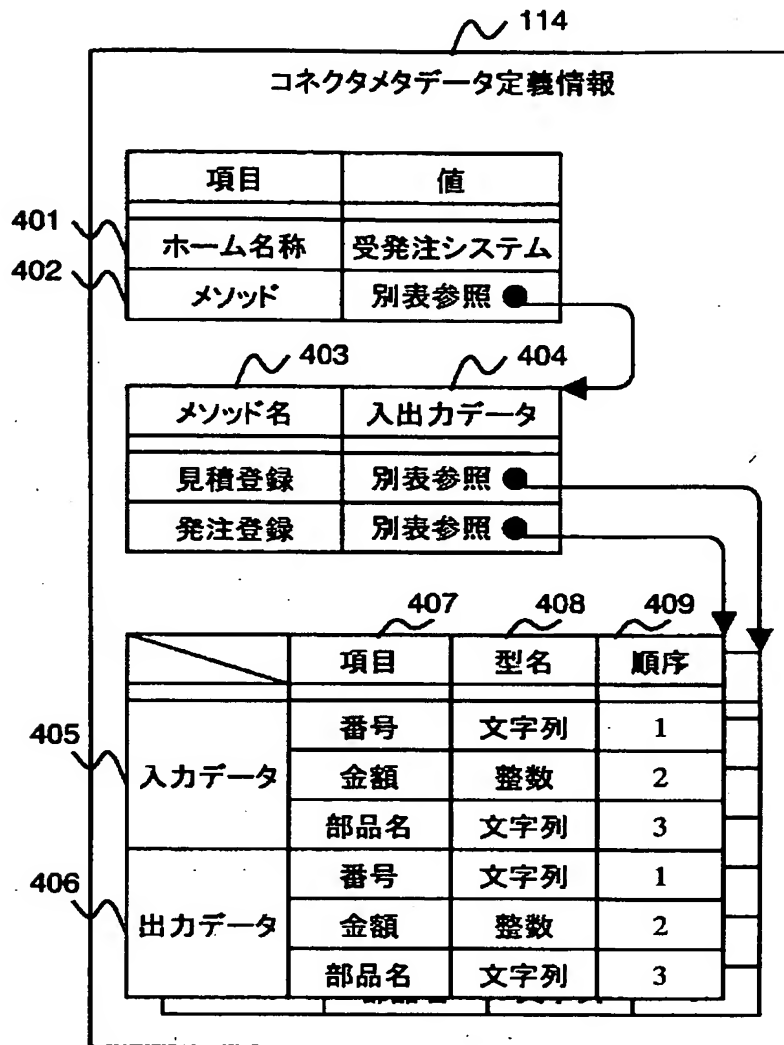
【図3】

図3

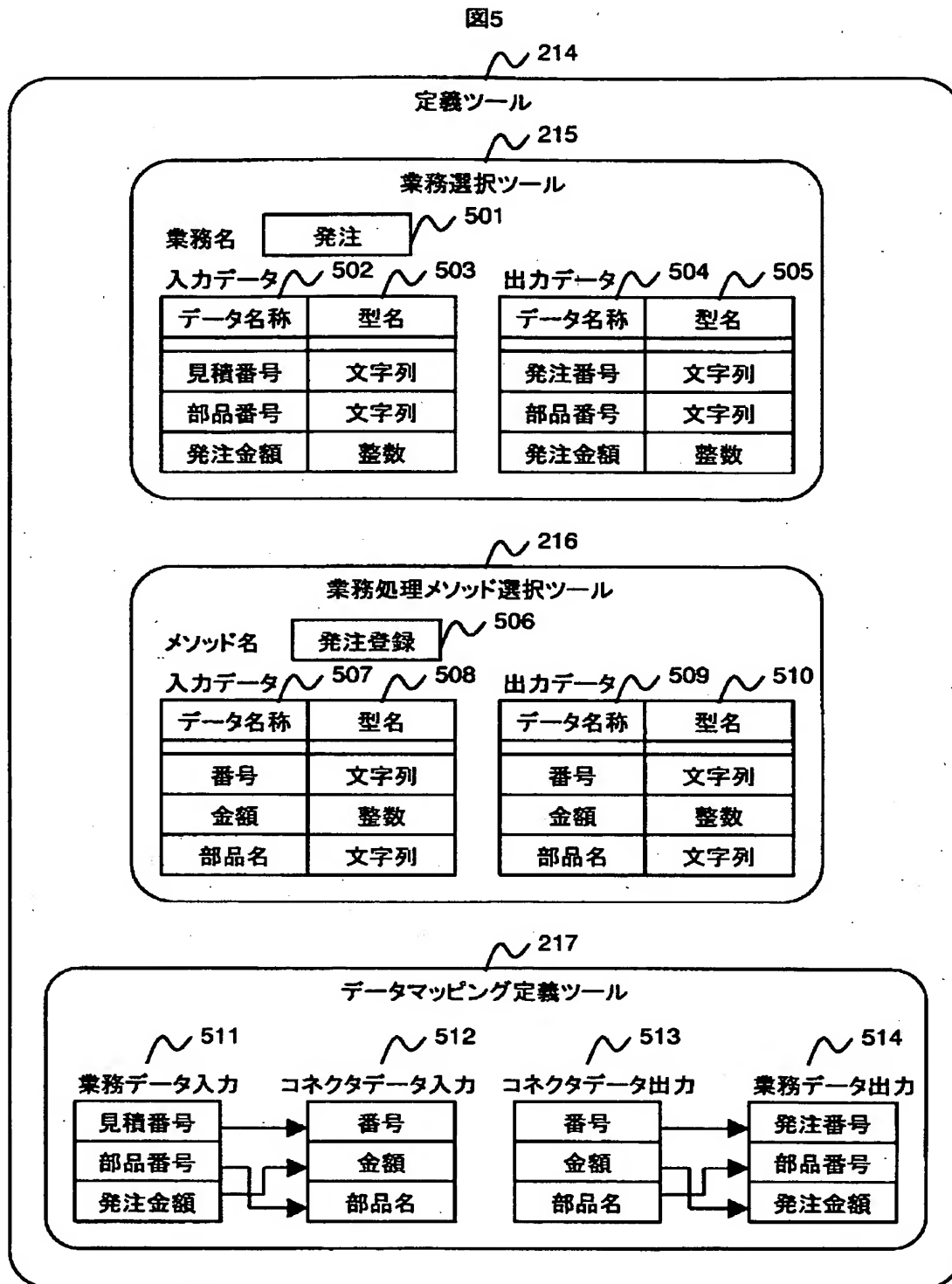


【図4】

図4

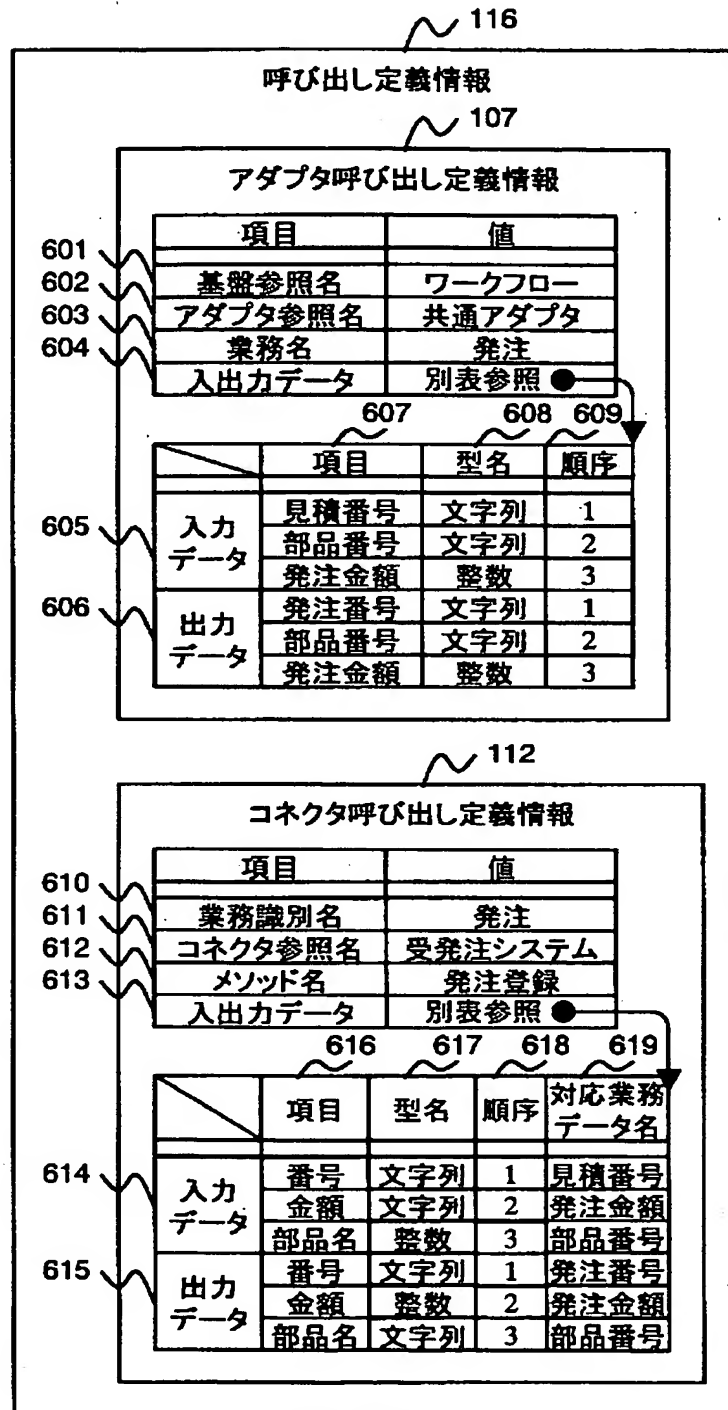


【図5】

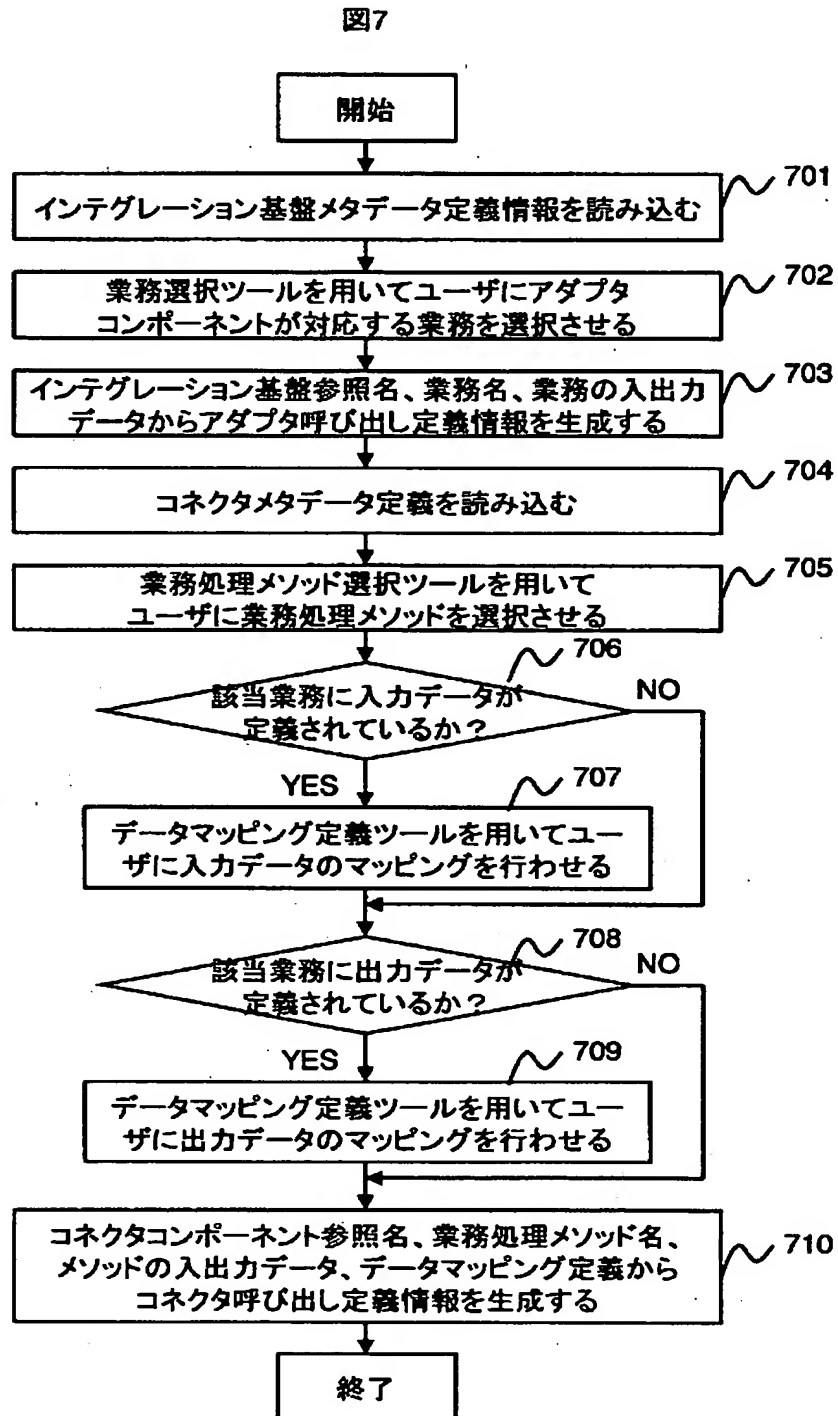


【図 6】

図6

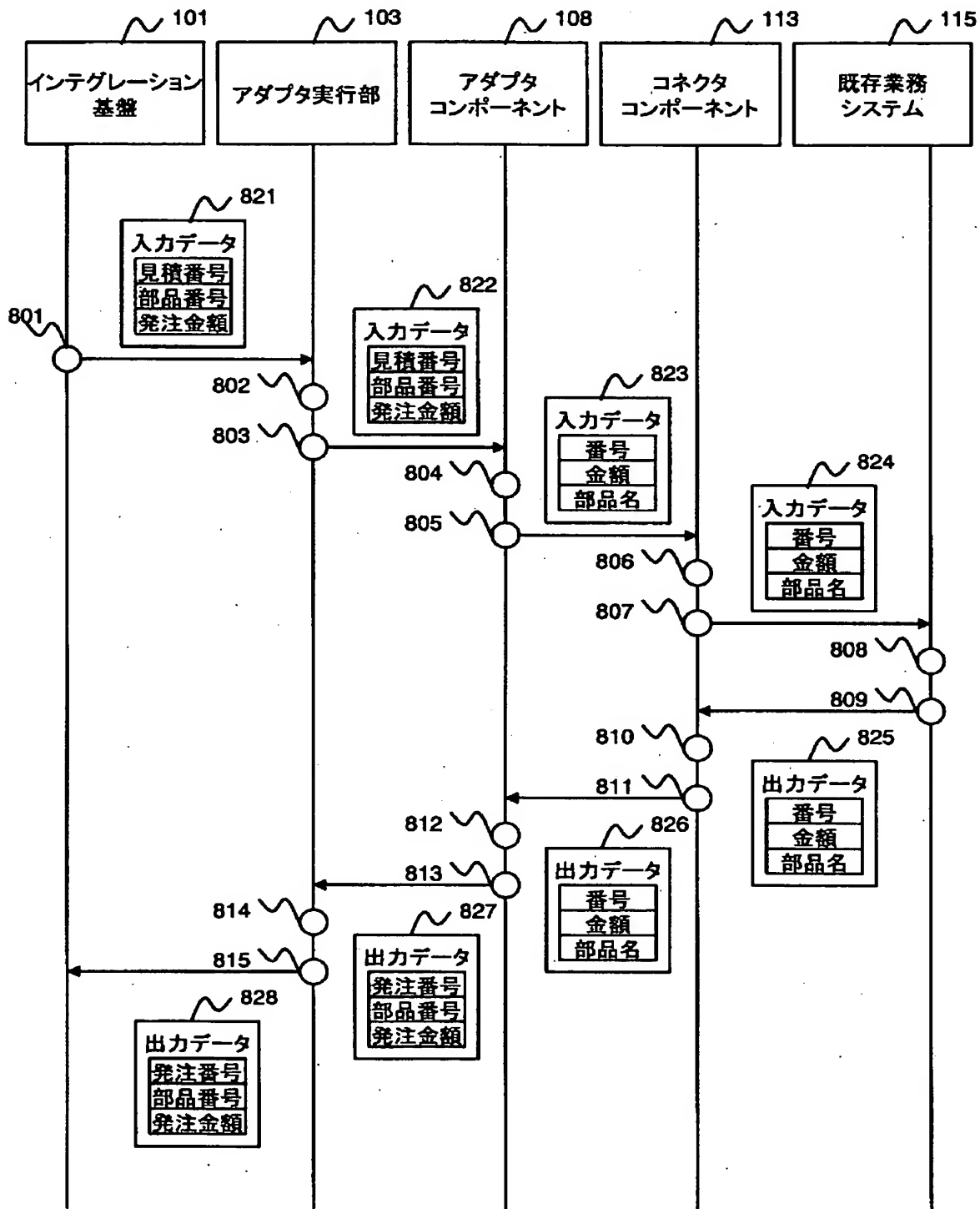


【図 7】



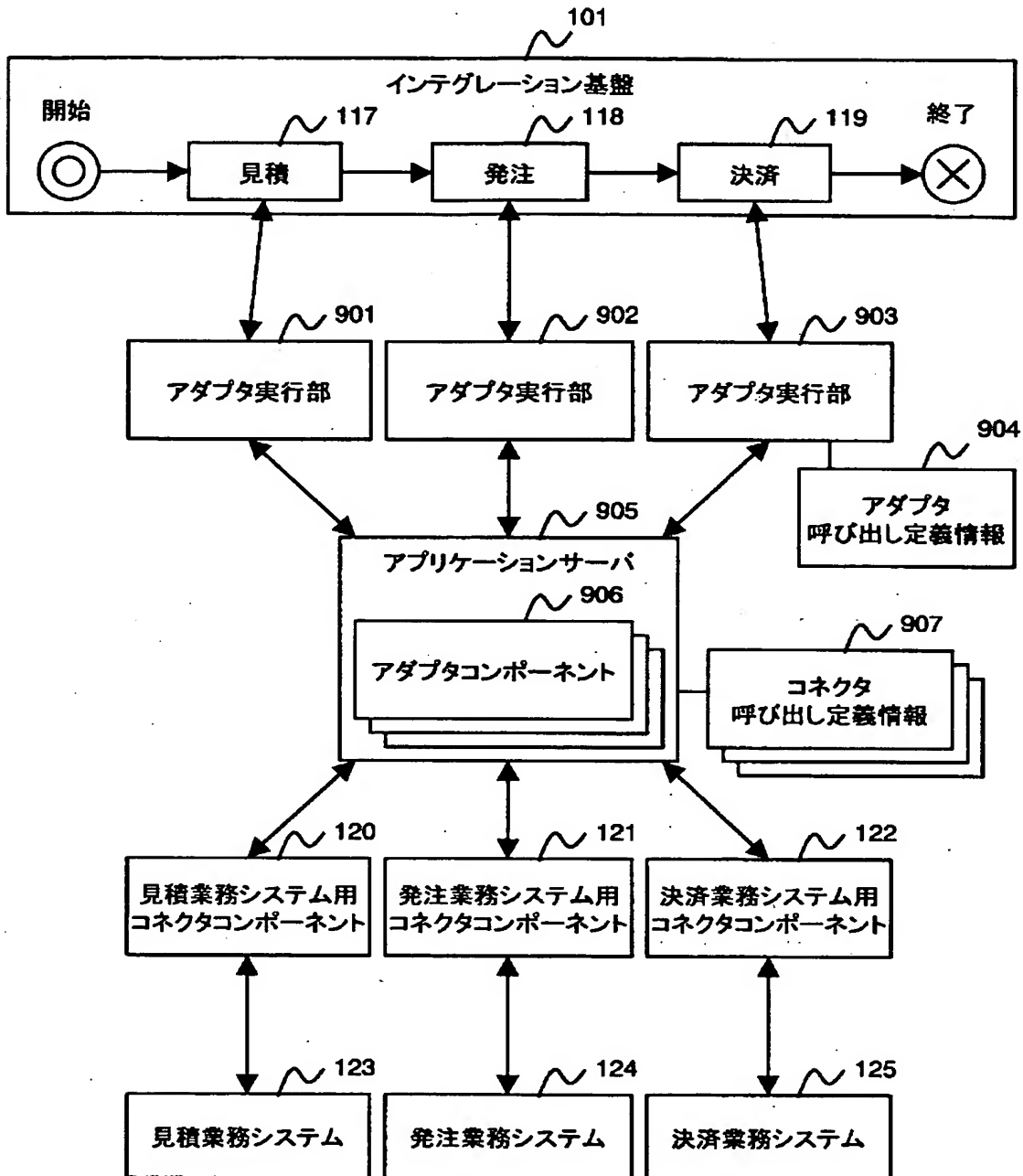
【図 8】

図8



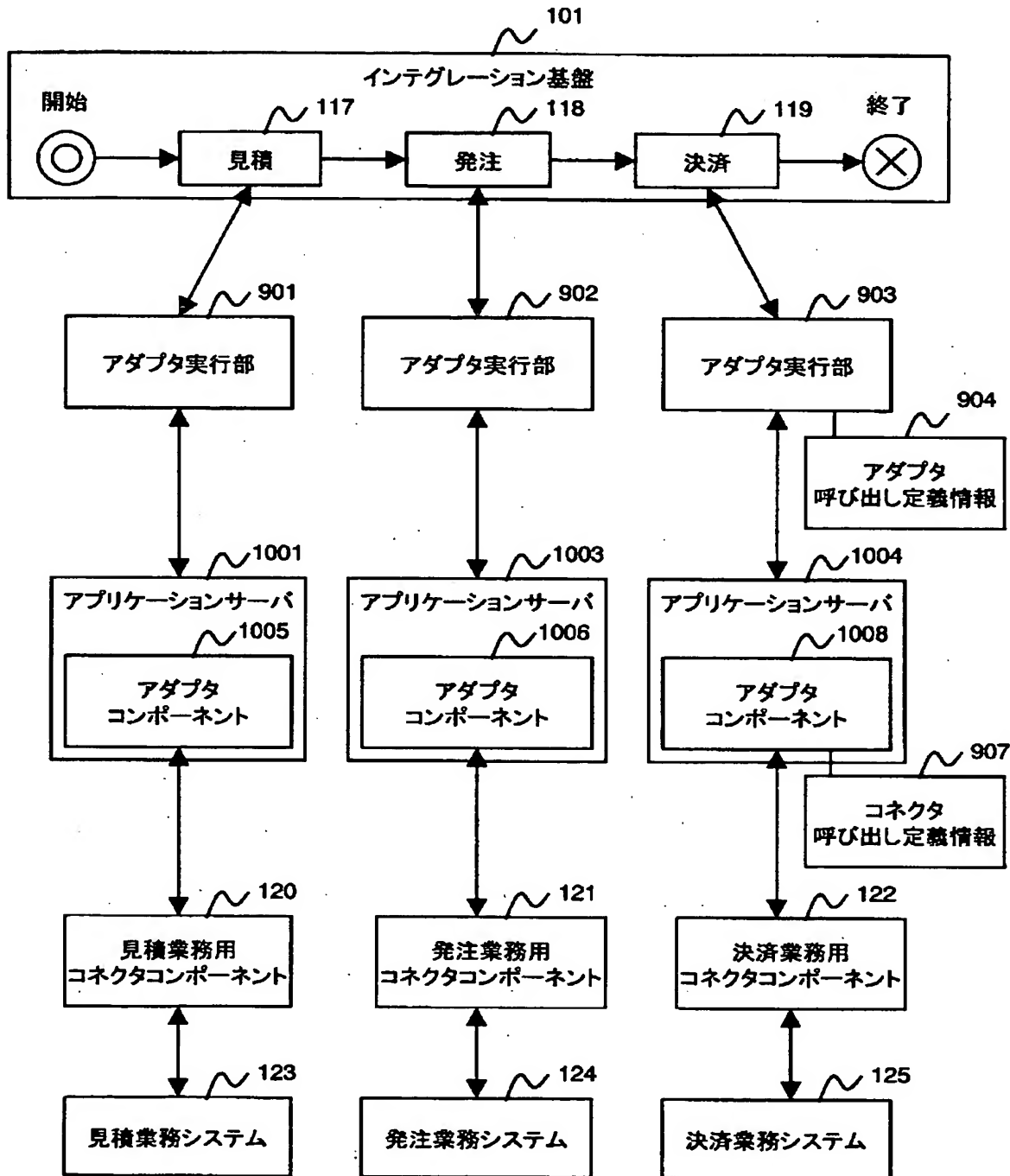
【図9】

図9



【図10】

図10



【書類名】 要約書

【要約】

【課題】

インテグレーション基盤とコネクタコンポーネントを接続するアダプタプログラムを開発する場合、アダプタプログラムを開発する者は、まずインテグレーション基盤との接続の状態を管理する手段、コネクタコンポーネントが提供しているサービスを利用する手段、アダプタコンポーネントのコンポーネントモデルを実装する手段、各コンポーネント間の複雑な通信インタフェースなどについて熟知している必要がある。

【解決手段】

コネクタコンポーネントに接続することができるアダプタコンポーネントと、インテグレーション基盤からの実行要求を受け付け、アダプタコンポーネントを起動するアダプタ実行部の二つをアダプタプログラムとして提供することにより、インテグレーション基盤から出力されたデータを入力として、コネクタコンポーネントの入力可能な形式に変換してコネクタコンポーネントに出力するプログラムの制御方法の特徴とする。

【選択図】 図1

特 2001-328505

認定・付加情報

特許出願の番号	特願2001-328505
受付番号	50101579969
書類名	特許願
担当官	第七担当上席 0096
作成日	平成13年10月29日

<認定情報・付加情報>

【提出日】	平成13年10月26日
-------	-------------

次頁無

出 願 人 履 歴 情 報

識別番号

[000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地

氏 名 株式会社日立製作所